

Low Complexity Decoding for Higher Order Punctured Trellis-Coded Modulation Over Intersymbol Interference Channels

Fabian Schuh and Johannes B. Huber

Institute for Information Transmission, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
mail: {schuh, huber}@LNT.de

Abstract—Trellis-coded modulation (TCM) is a power and bandwidth efficient digital transmission scheme which offers very low structural delay of the data stream. Classical TCM uses a signal constellation of twice the cardinality compared to an uncoded transmission with one bit of redundancy per PAM symbol, *i.e.*, application of codes with rates $\frac{n-1}{n}$ when 2^n denotes the cardinality of the signal constellation. Recently published work allows rate adjustment for TCM by means of puncturing the convolutional code (CC) on which a TCM scheme is based on. In this paper it is shown how punctured TCM-signals transmitted over intersymbol interference (ISI) channels can favorably be decoded. Significant complexity reductions at only minor performance loss can be achieved by means of reduced state sequence estimation.

Index Terms—trellis-coded modulation (TCM); punctured convolutional codes; Viterbi-Algorithm (VA); reduced state sequence estimation (RSSE); intersymbol interference (ISI);

I. INTRODUCTION

Ungerboeck's trellis-coded modulation (TCM) [1], [2] is an attractive digital transmission scheme when very low structural delay of the data stream is desired. Low structural latency is ensured by the use of convolutional codes instead of block codes (*cf.* [3]) and the dispense with interleaving (as opposed to convolutionally bit-interleaved coded modulation [4]). By expanding a constellation from 2^{n-1} to 2^n signal points and employing a rate- $\frac{n-1}{n}$ convolutional encoder one can improve the robustness of the transmission against noise by up to 6 dB without any further costs besides computational effort [1].

A recently published paper proposes to perform rate adjustment for TCM by means of puncturing the convolutional code (CC) on which a TCM scheme is based on. There, metric computations and trellis diagram become time-variant [5], [6].

Here, transmission over an ISI channel is considered which requires to apply a TCM-ISI super trellis for optimum decoding. As a result, the Viterbi algorithm (VA) has to be extended in order to handle both coded and uncoded bits in an optimal way (Please notice, that usual iterative equalization/decoding here is not possible due to lack of interleaver.).

II. SYSTEM MODEL

This paper deals with convolutionally encoded pulse-amplitude modulated (PAM) transmission as depicted in Fig. 1. Here, the term PAM is used for complex-valued signal constellations as well including amplitude-shift keying (ASK),

phase-shift keying (PSK) or quadrature-amplitude modulation (QAM) A binary data sequence $\langle u \rangle$ is split into n_u parallel

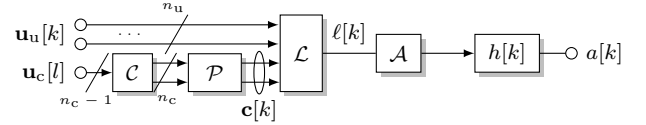


Fig. 1. System model for punctured trellis-coded modulation (P-TCM) with $n_u = 2$ and $n_c = 2$.

uncoded sequences $u_u[k]$ and $n_c - 1$ parallel sequences $u_c[l]$ that are encoded using a rate- $\frac{n_c-1}{n_c}$ binary convolutional encoder \mathcal{C} with generator polynomials $g_{ij}(D)$, $1 \leq i \leq n_c$; $1 \leq j \leq n_c - 1$ with max. degree ν , delay operator D , $n_c - 1$ parallel binary-input symbols and n_c parallel output symbols at each time instant.

At each output of the encoder, the symbols traverse through a puncturing system with puncturing scheme $\mathbf{P} = [P_{ij}]$, $P_{ij} \in \{0, 1\}$; $1 < i \leq n_c$; $1 < j < \Omega$ and period Ω . For each $(n_c - 1)$ -tuple of encoder input symbols the puncturing scheme cyclically advances by one step. Where P_{ij} is zero, the current symbol at the output is discarded, accordingly. The punctured n_c -ary encoded output symbols $c[k]$ together with the uncoded input symbols $u_u[l]$ form a label $\ell[k]$ by which the corresponding signal point out of $M = 2^{n_u+n_c}$ -ary constellation is selected. The transmit signal traverses through a dispersive discrete-time ISI channel modelled by a FIR-filter with memory L for $L + 1$ channel coefficients $h[k]$, $0 < k < L$ (*i.e.*, T -spaced sampling after the Whitened Matched Filter [7]).

III. TRELLIS-BASED DECODING

For sake of simplicity, we use for the following explanations a simple example, *i.e.*, a rate $\frac{1}{2}$ mother code (*e.g.*, $n_c = 1$) and a memory-1 ISI channel. Fig. 3 illustrates the extension of the trellis states by the uncoded symbols that are stored in the ISI channel. The two bars ($\overline{\quad}$) represent the generator polynomials g_1 and g_2 which generate the MSB and LSB of the (punctured) convolutional code (rate- $\frac{1}{2}$ mother code). This notation was introduced in [5], [6]. As the ISI channel stores the last L transmitted signal points, a super trellis needs to track not only the coded output of the (punctured) convolutional encoder but also the $L \cdot n_u$ last uncoded symbols. Thus, for each uncoded symbol and each channel tap (except for $h[0]$) an additional binary memory element must be added

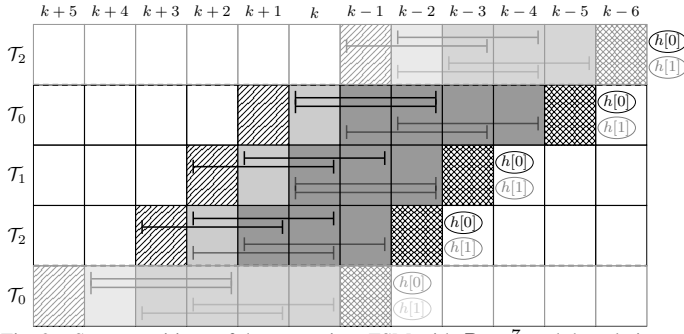


Fig. 2. State transitions of the transmitter FSM with $R = \frac{7}{3}$ and the relations between generator polynomials, FSM-state/input and channel state for a memory-1 ISI-channel with one uncoded bit stored in the states (crosshatched block).

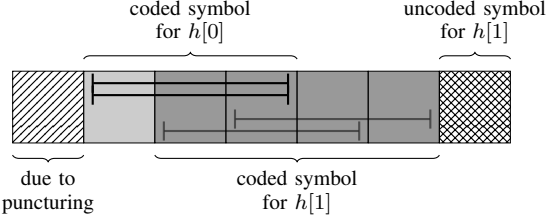


Fig. 3. Layout of a particular trellis state showing the individual components of the state ($n_u = 1$, $L = 1$, $\nu = 2$).

to the FSM, *e.g.*, for $n_u = 2$ and $L = 2$ a total of 4 additional memory elements have to be spent. As a result, the joint CC+ISI (super) trellis quickly becomes prohibitively large. However we will show later that reduced-state sequence-estimation (RSSE) techniques can be applied here, such that the computational complexity can be reduced significantly with only minor performance loss.

We first briefly consider an algorithm to construct the trellis of this finite state machine. In order to achieve optimal decoding we need to consider the coded and uncoded symbols in the finite-state machine as they take part in the memory of the CC and the ISI channel. However, as the uncoded symbols directly propagate to the mapper the uncoded symbols immediately affect the selection of the signal point and thus the input of the ISI channel.

To algorithmically handle the time-variant mapping we introduced [5] a set of so-called generator offsets \mathcal{T}_i which describe, depending on the puncturing scheme, modulation size, and time instant, the relations between generator polynomials, input value, FSM state, and mapping to MSB or LSB, respectively. For each new generator offset \mathcal{T}_i a new trellis segment arises, *e.g.*, the number of generator offsets equals the number of trellis segments in one trellis period.

The setup of a trellis representation of CC+ISI is defined by means of algorithm 1 in an abstract manner. The major steps are in line 10 and line 11 which fetches the uncoded symbols from the trellis state and appends it to the encoding results in order to perform the labeling. Then, $L + 1$ signal points are selected and weighted with the channel coefficients in line 13.

The routines `GETCURRENTSTATE` and `GETNEXTSTATE` have to evaluate the current and next state to construct the trellis as described in detail in [5], [6]. Additionally, we here also have to consider the uncoded symbols (`GETUNCODEDFROMSTATE`). The prototype routines for our

Algorithm 1 Building the FSM for P-TCM with $n_u > 0$ over ISI channels

Require: $\mathbf{h} \leftarrow \{h[0], h[1], h[2], \dots\}$

Require: Z number of states

```

1: for all  $s : 1 \rightarrow Z$  do
2:   for all  $\mathbf{u} \in \mathbb{F}_2^{n_c+n_u}$  do
3:     for all  $i : 1 \rightarrow \Lambda$  do
4:        $\mathbf{d} \leftarrow \{\mathbf{u}, s_0, s_1, \dots, s_{Z-1}\}$ 
5:        $\mathbf{s}^- \leftarrow \text{GETCURRENTSTATE}(\mathbf{d}, \mathcal{T}_i)$ 
6:        $\mathbf{s}^+ \leftarrow \text{GETNEXTSTATE}(\mathbf{d}, \mathcal{T}_i)$ 
7:       for  $\kappa : 0 \rightarrow (\text{length}(\mathbf{h}) - 1)$  do
8:          $\mathbf{e}(\kappa) \leftarrow \text{P-ENCODER}(\mathcal{T}_i, \kappa, \mathbf{g}_{\text{oct}}, \mathbf{d})$ 
9:       end for
10:       $\mathbf{u}_h \leftarrow \text{GETUNCODEDFROMSTATE}(\mathbf{u}, \mathcal{T}_i)$ 
11:       $\ell \leftarrow \text{LABELING}(\left\{ \mathbf{u}_h, \sum_{\kappa} \mathbf{e}(\kappa) \right\})$ 
12:       $\mathbf{m} \leftarrow \text{CONSTELLATION}(\ell)$ 
13:       $\mathbf{h}(\mathcal{T}_i, \mathbf{s}^-, \mathbf{u}) \leftarrow \mathbf{m}^\top \cdot \mathbf{h}$   $\triangleright$  hypotheses
14:       $\mathbf{T}(\mathcal{T}_i, \mathbf{s}^-, \mathbf{u}) \leftarrow \mathbf{s}^+$   $\triangleright$  next state
15:    end for
16:  end for
17: end for
18: function P-ENCODER( $\mathcal{T}_i, \kappa, \mathbf{g}_{\text{oct}}, \mathbf{d}$ )
19:   for  $i \in \{\text{MSB}, \text{LSB}\}$  do
20:      $\text{FIFO}_i \leftarrow$  shifts according to  $\mathcal{T}_i$ , and  $\kappa$ 
21:      $\mathbf{e}(i) \leftarrow \text{FIFO}_i^\top \cdot \mathbf{g}_{i, \text{dual}}$ 
22:   end for
23: end function

```

example ($\mathbf{P} = [(11)^\top (01)^\top]$, $\nu = 2$, $L = 1$) are given in algorithm 2 and return the content of the delay elements on the FIFO according to Fig. 2. Here, counting starts at the most left elements for each generator offset \mathcal{T}_i .

Algorithm 2 Prototype routines for the exemplary rate $\frac{7}{3}$ transmission scheme with a memory-2 convolutional code and the puncturing scheme $\mathbf{P} = [(11)^\top (01)^\top]$ (see Fig. 2. Additional $n_u = 1$ uncoded bits are used to address the signal points.

```

1: function GETCURRENTSTATE( $\mathbf{d}, \mathcal{T}_i$ )
2:   if  $i = 0$  then return  $\mathbf{d}(\{3, 4, 5, 6\})$ 
3:   else if  $i = 1$  then return  $\mathbf{d}(\{2, 3, 4, 5\})$ 
4:   else if  $i = 2$  then return  $\mathbf{d}(\{2, 3, 4, 5\})$ 
5:   end if
6: end function
7: function GETNEXTSTATE( $\mathbf{d}, \mathcal{T}_i$ )
8:   if  $i = 0$  then return  $\mathbf{d}(\{2, 3, 4\})$ 
9:   else if  $i = 1$  then return  $\mathbf{d}(\{1, 2, 3, 4\})$ 
10:  else if  $i = 2$  then return  $\mathbf{d}(\{1, 2, 3, 4\})$ 
11:  end if
12: end function

```

As a result, a matrix of hypotheses $\mathbf{h}(\mathcal{T}_i)$ and a matrix of state transitions $\mathbf{T}(\mathcal{T}_i)$ span a time-variant trellis with $2^{(n_u+n_c)}$

branches per state that can be used to optimally decode punctured TCM over ISI channels by means of a modified Viterbi algorithm. Here, \mathcal{T}_i denotes the i -th trellis segment out of the set of Λ segments. Hence, Λ is nominated as *trellis period*. The modifications necessary for the Viterbi algorithm comprise two separate path registers for uncoded and coded symbols. This is due to the differing symbol phases w.r.t. the puncturing matrix, *e.g.*, differing data speed of \mathbf{u}_u and \mathbf{u}_c (*cf.*, Fig. 1). The metric computations are described in the next section.

IV. REDUCED-STATE SEQUENCE ESTIMATION

We will now describe the application of RSSE by recalling the basic principles of Delayed Decision Feedback Sequence Estimation [8] (DFSE). We will briefly show the metric calculations before giving numerical simulation results.

A. State Reduction Techniques (PAM without channel code)

1) *DFSE*: When equalizing uncoded digital PAM signaling over a discrete-time ISI channel with $L+1$ taps using *delayed decision feedback sequence estimation* (DFSE), a trellis is constructed from the first $\tilde{L} \leq L$ taps only. Thus, the number of states is reduced from M^L to $M^{\tilde{L}}$. For $\tilde{L} = L$ DFSE is equivalent to MLSE decoding via full-state VA. In contrast, when $\tilde{L} = 0$, the resulting trellis has a single state and, thus, represents a decision feedback equalization (DFE). Hence, DFSE allows an efficient way to trade between full-state VA and one-state DFE [8]. The remaining $L+1-\tilde{L}$ channel taps are considered in a delayed decision-feedback equalization (DFE) that is performed in each trellis state using the *delayed* path register of the corresponding state.

The main difference to full state equalization appears in the metric computation for each time instant. From eq. (1) it can be seen that the state specific path register $p_{\text{reg}}[k, s]$ is delayed by \tilde{L} and its elements are multiplied by the subsequent channel coefficients $h_{\text{dfe}}[h]$ which have not been considered in the trellis. The branch metric $\lambda(s, u)$ (*e.g.*, Euclidean distance of the received symbol $y[k]$ to the hypotheses $h(s, u)$ for the state s and symbols u) thus includes a DFE-term δ :

$$\delta = \sum_{\kappa} p_{\text{reg}}[k - \tilde{L} + \kappa, s] \cdot h_{\text{dfe}}[\kappa] \quad (1)$$

$$\lambda(s, u) = |y[k] - h(s, u) - \delta|^2$$

2) *RSSE*: In RSSE, on the other hand, Z arbitrary MLSE states, each with $2^{n_c+n_u-1} = \frac{M}{2}$ possible branches to adjacent states, are combined into $Z_R = \frac{Z}{2^J}$; $J \in \mathbb{N}$ *hyperstates* [9], [10] each having 2^J substates and $2^K \cdot 2^J$ branches. A certain assignment of states to hyperstates is called *partitioning* [10].

Instead of having 2^K arriving branches at each of the Z MLSE states we get a set of $2^K \cdot 2^J$ branches at each of the Z_R hyperstates. The total number of available branches remains $2^K \cdot Z$. However, when using RSSE only 2^K branches are possible (*i.e.*, enabled) from each state, at a given time instant. The availability of branches is determined by the path register in each state, and is thus a form of decision-feedback.

The main difference of DFSE and RSSE to MLSE is, that we decide for a surviving path prematurely resulting in a

truncation of error events. A loss in Euclidean distance appears if an error event with minimum Euclidean distance gets truncated. Therefore the performance of RSSE strongly depends on the partitioning of the states into hyperstates. However, survivor-decision specifies the sub-state within a hyperstate uniquely and thus allows correct metric calculation. Instead of exhaustively search for the optimum state partitioning, which maximizes the intra-hyperstate distance [10], we exploit the minimum phase characteristics of the ISI channel which is, as described above, fully integrated into our trellis.

For a minimum phase channel impulse response the prior channel input symbols are weighted fewer than more recent once and, thus, affect the metric less. Hence, the intra-hyperstate distance is maximized when states are combined with respect to elder positions in the state number. This particular partitioning is equivalent to DFSE for ISI channels and will be called *DFSE partitioning*. The minimum phase ISI channel is the last element to affect the received symbols and is also fully integrated into the FSM. Thus we can apply the *DFSE partitioning* to use RSSE for punctured TCM (P-TCM) over ISI channels.

B. Implementation Issues

Algorithm 3 Metric calculations for total RSSE – J^{th} partition

```

1:  $q_c \leftarrow \log_2(\text{nr. hyperstates})$ 
2: for all  $s \in \mathcal{S}$  do
3:   for all  $u \in \mathcal{A} \cdot K$  do
4:     for all  $\kappa = 0 \rightarrow J-1$  do ▷ active branches
5:       if  $\kappa \leq L$  then
6:          $\zeta[\kappa] = p_{\text{reg,uncoded}}(L_{\text{traceback}} - L + \kappa, s)$ 
7:       else
8:          $\zeta[\kappa] = p_{\text{reg,coded}}(L_{\text{traceback}} - L - q_c + \kappa, s)$ 
9:       end if
10:    end for
11:     $\lambda(\zeta, u) \leftarrow \Gamma(\zeta) + |y[k] - h(\zeta, u)|^2$ 
12:  end for
13: end for
```

Most importantly, in terms of implementation, we need to distinguish between two path registers, namely one for the uncoded symbols $p_{\text{reg,uncoded}}$, that pass through the ISI channel only, and one for the coded symbols $p_{\text{reg,coded}}$. Hence, when performing the feedback in RSSE both registers need to be considered, depending on the partition depth J . Algorithm 3 shows the application of both path registers in order to select the available paths by means of an indicator variable ζ which specifies the active branches by means of decision-feedback.

C. Numerical Results

Numerical result of a computer simulation of a transmission scheme with the following parameters are given. The total transmission rate is $\frac{7}{3}$ with a memory-2 convolutional code with generator polynomials $[7, 3]_8$ and the puncturing scheme $\mathbf{P} = \begin{bmatrix} (11)^T & (01)^T \end{bmatrix}$. To increase the transmission rate from $\frac{4}{3}$ to $\frac{7}{3}$ additional $n_u = 1$ uncoded bits are used to

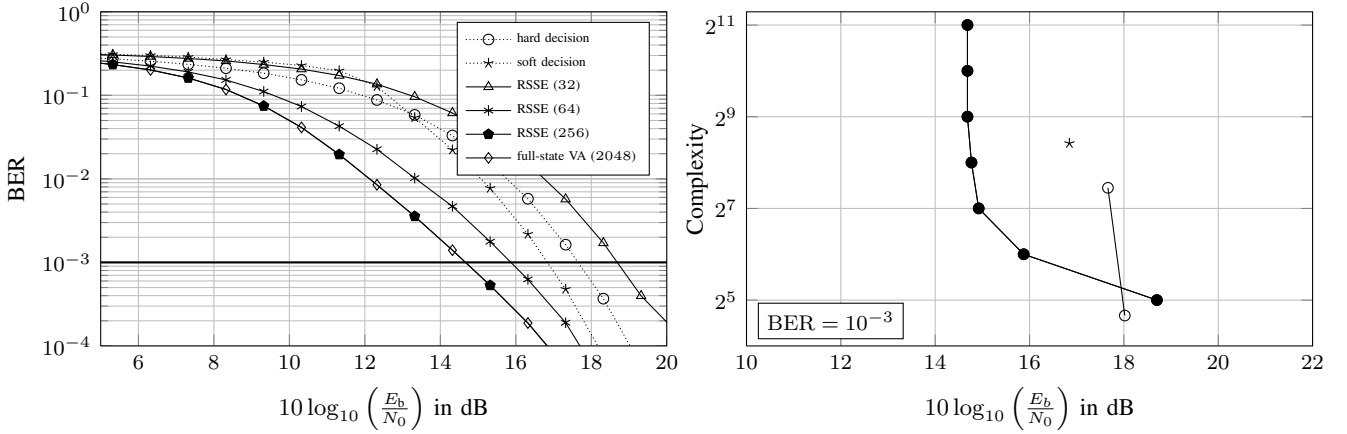


Fig. 4. BER Performance and computational decoder complexity for a rate $\frac{7}{8}$ transmission scheme with a memory-2 convolutional code $[(7, 3)_8]$ and the puncturing scheme $\mathbf{P} = [(11)^T (01)^T]$. Additional $n_u = 1$ uncoded bits are used to address the signal points and the ISI channel has memory $L = 2$.

address a signal points from an 8-ASK constellation (cf., Fig. 5). Additionally, square QAM constellations may be built from two independent ASK constellations in in-phase and quadrature component to further increase transmission rate without increasing the signal bandwidth. Thus, for all usual PAM constellations P-TCM is favorably based on the one-dimensional 4-ASK constellation because fine tuning of the transmission rate is possible by means of puncturing and addition of uncoded bits. The transmit signal traverses through an ISI channel $h[\kappa]$ with memory $L = 2$ plus additive white Gaussian noise.

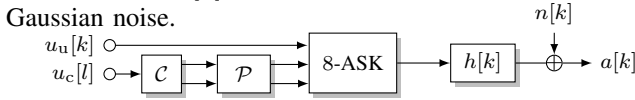


Fig. 5. System model for punctured trellis-coded modulation (P-TCM) with $n_u = 1$ and $n_c = 2$.

We use the following unit energy channel $h[\kappa]$:

$$h_{\text{lin}}[\kappa] = \frac{L - \kappa + 1}{L + 1} \quad \text{for } 0 \leq \kappa \leq L$$

$$h[\kappa] = \frac{1}{\sqrt{\sum_{\gamma} |h_{\text{lin}}[\gamma]|^2}} h_{\text{lin}}[\kappa]$$

Figure 4 shows bit error rates versus $\frac{E_b}{N_0}$ from simulations. We compare the results of our proposed approach with separated equalization and decoding using BCJR and DFSE for soft-/hard equalization of the ISI and a full-state VA for decoding of the CC. Note that, due to the objective of very low structural delay of the data stream no interleaver is applicable and thus no iterative equalization-decoding procedure is possible. Clearly, the proposed scheme outperforms the separated approaches by several decibels. However, the full-state trellis complexity number¹ is 2048 and thus significantly higher when compared to the separated approaches. Hence we reduce the computational complexity by sacrificing optimality by reduced-state sequence estimation (RSSE). By this, we can reduce the complexity number from 2048 down to 256 without noticeable loss in performance. When further reducing to a complexity number of 64 the performance is still

¹The computation trellis complexity number is defined as the number of metric calculations per information bit.

slightly better as the soft-equalization and decoding approach, although the latter has a complexity number of roughly 1369. Computational complexity¹ over $\frac{E_b}{N_0}$ that is required to achieve a bit error probability of less than 10^{-3} is shown in Fig. 5. Obviously our approach allows to reduce the complexity to 2^6 and still outperforms the separate soft-equalization/decoding approach.

V. CONCLUSION

It has been shown that maximum-likelihood decoding for punctured TCM can be achieved with low computational complexity and very good performance even for transmission over ISI channels. Thus, punctured TCM can be applied as a low-latency transmission scheme with high spectral efficiency. An additional benefit is the improved flexibility in transmission rate due to the flexible choice of the puncturing scheme, in contrast to classical TCM which can only achieve integer rates.

REFERENCES

- [1] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. IT-28, no. 1, pp. 55 – 67, jan 1982.
- [2] —, "Trellis-coded modulation with redundant signal sets; part i: Introduction; part ii: State of the art," *Communications Magazine, IEEE*, vol. 25, no. 2, pp. 5 –21, February 1987.
- [3] T. Hehn and J. B. Huber, "LDPC Codes and Convolutional Codes with Equal Structural Delay: A Comparison," *IEEE Transactions on Communications*, vol. 57, no. 6, pp. 1683–1692, June 2009. [Online]. Available: http://www.lit.lnt.de/papers/tr_com_2009_hehn.pdf
- [4] E. Zehavi, "8-PSK Trellis Codes for a Rayleigh Channel," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 873 –884, may 1992.
- [5] F. Schuh and J. Huber, "Low Complexity Decoding for Punctured Trellis-Coded Modulation over Intersymbol Interference Channels," in *accepted for 2014 International Zurich Seminar on Communications*, Feb. 2014.
- [6] F. Schuh, A. Schenk, and J. Huber, "Punctured Trellis-Coded Modulation," in *submitted to ICC 2014*, Jun. 2014.
- [7] J. Forney, G., "Maximum-likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. Inf. Theory*, vol. 18, no. 3, pp. 363–378, 1972.
- [8] A. Duel-Hallen and C. Heegard, "Delayed Decision-Feedback Sequence Estimation," *IEEE Trans. Commun.*, vol. 37, no. 5, pp. 428–436, May 1989.
- [9] J. Huber, *Trelliscodierung: Grundlagen und Anwendungen in der digitalen Übertragungstechnik*. Springer, 1992.
- [10] B. Spinnler and J. Huber, "Design of Hyper States for Reduced-State Sequence Estimation," in *Proc. IEEE Int. Conf. Communications ICC '95 Seattle*, vol. 1, 1995, pp. 1–6.